



Large Scale Linear Programs and Heuristics for the Design of Survivable Telecommunication Networks

IRADJ OUVEYSI, ANDREW WIRTH* and ANNIE YEH
The University of Melbourne, VIC 3010, Australia

wirth@mame.mu.oz.au

OSMAN OĞUZ
Bilkent University, Ankara, Turkey

Abstract. We consider the problem of providing a minimum cost multi-service network subject to one link failure scenarios. We assume our network is fully meshed and demand is satisfied by using direct or two hop-paths. We provide a large scale linear programming formulation and propose and test two effective heuristics.

Keywords: survivable, link failure

In the light of new technologies in the telecommunications industry, it is possible to accommodate multiple services in the same transport system Medhi (1991). Thus we need to provide new tools to design networks which will be able to handle such services. Here we address the problem of dimensioning of a telecommunication network subject to some survivability constraints and alternative routing schemes Ash, Cardwell, and Murray (1981), Cameron et al. (1983), Gibbens and Key (1988, 1990), Gibbens and Kelly, 1990). We assume that the network is subject to a single link failure Ouveysi and Tham (1994) and consider these survivability issues in the design process of a multi-service traffic network.

We can construct a minimum-cost network to satisfy the multi-services traffic matrices, assuming that idle capacity in transmission links may be re-deployed in forming legs of two-hop paths by dynamic routing schemes. Throughout this work we consider a fully meshed network $G = \langle V, E \rangle$ ($|E| = m = n(n - 1)/2$), where V is the set of nodes and E is the set of edges (transmission links). Let (r_{ij}^s) represent the symmetric traffic flow requirement matrix for all node pairs in G for service type $s \in S = \{1, 2, \dots, N_s\}$ ($|S| = N_s$).

Our objective is to design a minimum-cost network to satisfy traffic flow requirements for all services in the case of any single link failure. We assume that the traffic flow between a pair of nodes is satisfied by using direct or two-hop paths. By e_{ij} we mean the link between two nodes $i, j \in V$ and let the cost of one unit of link capacity between two nodes $i, j \in V$ be $c_{ij} > 0$.

* Corresponding author: Department of Mechanical and Manufacturing Engineering, The University of Melbourne, VIC 3010, Australia.

1. Mathematical modelling

1.1. Multi-service scenario with no link failure

Here we assume that no link in the network is subject to failure and model the multi-service traffic design problem as a minimum-cost network flow problem. As input to our network design problem, the multi-service traffic profile is given as a number of end-to-end traffic matrices. We assume that the amount of the flow of any service on any link is bounded above to satisfy a grade of service for any service type. Let us define the following parameters and variables:

U_{ij}^s =: The upper bound for flow on link e_{ij} for service $s \in S$, $\forall i < j \in V$.

y_{ij} =: The total amount of capacity assigned to link e_{ij} , $\forall i < j \in V$.

x_{ij}^s =: The direct flow between nodes i and j along the e_{ij} , $\forall i < j \in V$, $\forall s \in S$.

x_{ikj}^s =: The flow between nodes i and j , through the tandem node k along the links e_{ik} and e_{kj} , $\forall i < j \in V$, $\forall s \in S$.

For any service type s and for any origin–destination pair the traffic flow requirement constraint is as

$$x_{ij}^s + \sum_{l \in V, l \neq i, j} x_{ilj}^s \geq r_{ij}^s \quad \forall i < j, \forall s \in S. \quad (1)$$

The capacity requirement constraints are

$$y_{ij} - \sum_{s \in S} \left(x_{ij}^s + \sum_{l \in V, l \neq j, l > i} x_{ijl}^s + \sum_{l \in V, l > j} x_{jil}^s + \sum_{l \in V, l \neq i, l < j} x_{lij}^s + \sum_{l \in V, l < i} x_{lji}^s \right) \geq 0 \quad (2)$$

$$x_{ij}^s + \sum_{l \in V, l \neq j, l > i} x_{ijl}^s + \sum_{l \in V, l > j} x_{jil}^s + \sum_{l \in V, l \neq i, l < j} x_{lij}^s + \sum_{l \in V, l < i} x_{lji}^s - U_{ij}^s \leq 0 \quad (3)$$

$$x_{ij}^s, x_{ilj}^s, y_{ij} \geq 0 \quad \forall i < j \in V, \forall l \in V, l \neq i, j. \quad (4)$$

The total number of x_{ij}^s and x_{ilj}^s variables in a traffic flow requirement constraint is equal to $n - 1$. For any service, we have m constraints of type (1), hence the number of x variables is equal to

$$N_s m(n - 1)$$

and since the number of y variables is m , then total number of variables is equal to

$$N_s m(n - 1) + m = N_s \frac{n(n - 1)}{2} (n - 1) + \frac{n(n - 1)}{2} = \frac{n(n - 1)}{2} (N_s(n - 1) + 1).$$

For any service, we have m constraints of type (1) and m constraints of type (3). Since the number of constraints of type (2) is equal to m , hence the total number of constraints in the non-failure case is equal to

$$2N_s m + m = \frac{n(n-1)}{2}(2N_s + 1).$$

Then the multi-service network problem (with no link failure case) can be formulated as the following linear programming problem:

$$\begin{aligned} \text{(P0)} \quad & \min \sum_i \sum_{j>i} c_{ij} y_{ij} \\ & \text{subject to constraints (1)–(4).} \end{aligned}$$

The total number of variables and constraints of (P0) are in the order of $N_s n^3$ and $N_s n^2$, respectively.

1.2. Multi-service traffic design with protection against one link failure

Here we consider a multi-service traffic design as in section 1.1 with the further assumption that any one link in the network is subject to failure. Since the total number of the links of the network for the fully meshed case is equal to m , we have m different failure scenarios. Here we need to define the following variables.

$x_{ij}^{s,p}$ =: The direct flow between nodes i and j along the link e_{ij} for service $s \in S$ for scenario p .

$x_{ikj}^{s,p}$ =: The flow between nodes i and j , through the tandem node k along the links e_{ik} and e_{kj} for service $s \in S$ for scenario p .

We also define the following binary indicators:

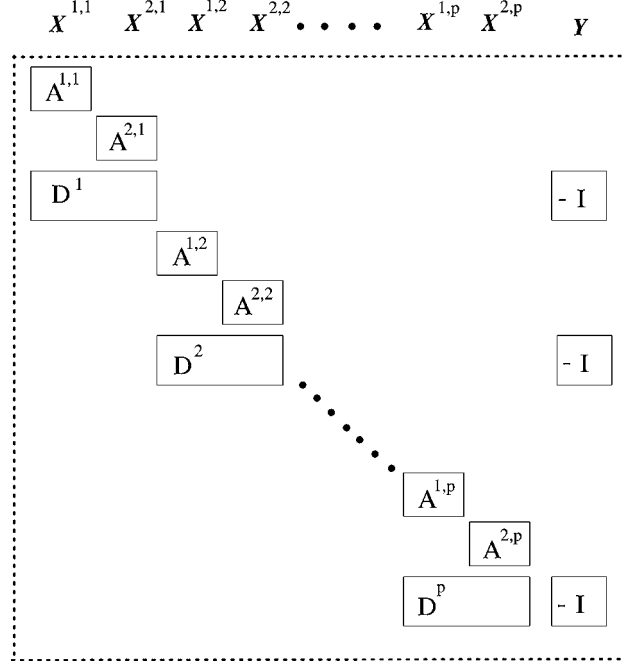
$$\begin{aligned} \delta_{ij}^p &= \begin{cases} 0 & \text{if } e_{ij} \text{ is the failed link for scenario number } p, \\ 1 & \text{otherwise;} \end{cases} \\ \Delta_{ilj}^p &= \delta_{il}^p \cdot \delta_{lj}^p. \end{aligned}$$

For any service type $s \in S$ and any failure scenario p , the traffic flow requirement constraints are

$$x_{ij}^{s,p} \cdot \delta_{ij}^p + \sum_{k \in V, k \neq i, j} x_{ikj}^{s,p} \cdot \Delta_{ikj}^p \geq r_{ij}^s \quad \forall i < j \in V, \forall p, \forall s \in S. \quad (5)$$

The capacity requirement constraints are

$$\begin{aligned} y_{ij} - \sum_{s \in S} \left(x_{ij}^{s,p} \cdot \delta_{ij}^p + \sum_{l \in V, l \neq j, l > i} x_{ijl}^{s,p} \cdot \Delta_{ijl}^p + \sum_{l \in V, l > j} x_{jil}^{s,p} \cdot \Delta_{jil}^p \right. \\ \left. + \sum_{l \in V, l \neq i, l < j} x_{lij}^{s,p} \cdot \Delta_{lij}^p + \sum_{l \in V, l < i} x_{lji}^{s,p} \cdot \Delta_{lji}^p \right) \geq 0 \quad \forall i < j \in V, \forall p, \end{aligned} \quad (6)$$

Figure 1. Structure of the constraint matrix for two services and p scenarios.

$$\begin{aligned}
 & x_{ij}^{s,p} \cdot \delta_{ij}^p + \sum_{l \in V, l \neq j, l > i} x_{ijl}^{s,p} \cdot \Delta_{ijl}^p + \sum_{l \in V, l > j} x_{jil}^{s,p} \cdot \Delta_{jil}^p + \sum_{l \in V, l \neq i, l < j} x_{lij}^{s,p} \cdot \Delta_{lij}^p \\
 & + \sum_{l \in V, l < i} x_{lji}^{s,p} \cdot \Delta_{lji}^p - U_{ij}^s \leq 0 \quad \forall i < j \in V, \forall p, \forall s \in S, \quad (7)
 \end{aligned}$$

$$x_{ij}^{s,p}, x_{ilj}^{s,p}, y_{ij} \geq 0 \quad \forall i < j \in V, \forall l \in V, l \neq i, j. \quad (8)$$

1.2.1. Matrix representation

The constraint matrix is shown in figure 1 for the case $N_s = 2$. Here $A^{s,p}$ represents the constraint matrix (left-hand side) corresponding to the constraints of type (5) and (7) for service type s and scenario p . Matrix D^p together with matrix I represent the constraint matrix of the constraints of type (6) for scenario p , where I denotes an $m \times m$ identity matrix. As seen from the structure of the constraint matrix, our problem is similar to the non-simultaneous multi-commodity problem. In the following sections we will show that the size of this matrix is in the order of $N_s n^4 \times N_s n^5$ and hence solving this linear programming model for large problems is almost impossible. Thus we will present a heuristic for solving this problem.

1.2.2. Number of variables

We compute the number of $x_{ijl}^{s,p}$ variables by first summing by il then by j and finally by p , to obtain $N_s |E| (n-2) (|E|-2)$. The number of $x_{ij}^{s,p}$ variables is $N_s |E| (|E|-1)$,

obtained by summing by ij then by p . So the total number of x variables is

$$\begin{aligned} & N_s \frac{n(n-1)}{2} (n-2) \left\{ \frac{n(n-1)}{2} - 2 \right\} + N_s \frac{n(n-1)}{2} \left\{ \frac{n(n-1)}{2} - 1 \right\} \\ &= N_s \frac{n(n-1)(n-2)(n^2-3)}{4}. \end{aligned}$$

The total number of y variables is equal to $|E| = n(n-1)/2$. So the total number of variables is

$$N_s \frac{n(n-1)(n-2)(n^2-3)}{4} + \frac{n(n-1)}{2}.$$

1.2.3. Number of constraints

For any service $s \in S$ and for each scenario p we have m constraints of type (5) and $m-1$ constraints of type (7). The number of constraints of type (6) for each scenario p is equal to $m-1$. Hence the total number of constraints is

$$\begin{aligned} TNC &= N_s m^2 + N_s m(m-1) + m(m-1) \\ &= m[N_s(2m-1) + m-1]. \end{aligned}$$

Thus our multi-service network problem (subject to single link failure) can be modelled as the following linear programming problem:

$$\begin{aligned} \text{(MS)} \quad & \min \sum_i \sum_{j>i} c_{ij} y_{ij} \\ & \text{subject to constraints (5)–(8).} \end{aligned}$$

The total number of variables of problem (MS) is in the order of $N_s n^5$ and the number of its constraints is in the order of $N_s n^4$.

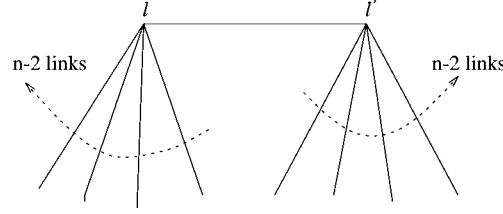
2. Two heuristics

Given a fully meshed graph $G = \langle V, E \rangle$, with $N_s \in \mathbb{Z}^+$ different service types and a symmetric origin–destination demand matrix (r_{ij}^s) , $i, j \in V$, we need to find the capacities of the links of G such that total cost of capacity allocation is minimized while all the origin–destination demands are satisfied in case of any one link failure in the network.

2.1. A linear programming heuristic

For any failure scenario p , we can provide a linear programming model which represents this failure scenario as follows:

$$\text{(MP)} \quad \min \sum_i \sum_{j>i} c_{ij} y_{ij}$$

Figure 2. The paths that are affected by the failure of link $e_{ll'}$.

subject to

$$x_{ij}^s \cdot \delta_{ij}^p + \sum_{k \in V, k \neq i, j} x_{ikj}^s \cdot \Delta_{ikj}^p \geq r_{ij}^s \quad \forall i < j \in V, \forall s \in S. \quad (9)$$

The capacity requirement constraints are

$$y_{ij} + \bar{y}_{ij} - \sum_{s \in S} \left(x_{ij}^s \cdot \delta_{ij}^p + \sum_{l \in V, l \neq j, l > i} x_{ijl}^s \cdot \Delta_{ijl}^p + \sum_{l \in V, l > j} x_{jil}^s \cdot \Delta_{jil}^p + \sum_{l \in V, l \neq i, l < j} x_{lij}^s \cdot \Delta_{lij}^p + \sum_{l \in V, l < i} x_{lji}^s \cdot \Delta_{lji}^p \right) \geq 0 \quad \forall i < j \in V, \quad (10)$$

$$x_{ij}^s \cdot \delta_{ij}^p + \sum_{l \in V, l \neq j, l > i} x_{ijl}^s \cdot \Delta_{ijl}^p + \sum_{l \in V, l > j} x_{jil}^s \cdot \Delta_{jil}^p + \sum_{l \in V, l \neq i, l < j} x_{lij}^s \cdot \Delta_{lij}^p + \sum_{l \in V, l < i} x_{lji}^s \cdot \Delta_{lji}^p - U_{ij}^s \leq 0 \quad \forall i < j \in V, \forall s \in S, \quad (11)$$

$$x_{ij}^s, x_{ilj}^s, y_{ij} \geq 0 \quad \forall i < j \in V, \forall l \in V, l \neq i, j. \quad (12)$$

Here (\bar{y}_{ij}) are non-negative constants and may be interpreted as existing link capacities. (MP) determines a minimum cost augmentation of link capacities to satisfy traffic flow requirements (r_{ij}^s) , $s \in S$ for any failure scenario p .

For any service type $s \in S$ we have m requirement constraints of type (9), of which $2n - 3$ have $n - 2$ variables of type x each (see figure 2) and the remaining $m - (2n - 3)$ constraints with $n - 1$ variables. Hence total number of variables is equal to

$$N_s [(2n - 3)(n - 2) + (m - 2n + 3)(n - 1)] + m.$$

We also have $m - 1$ constraints of type (10) and $N_s(m - 1)$ constraints of type (11). The total number of variables is thus

$$N_s m + (m - 1) + N_s(m - 1) = N_s(2m - 1) + m - 1.$$

This shows that the total number of constraints of (MP) is in the order of $N_s n^2$ and the total number of its variables is in the order of $N_s n^3$.

For the network G consider the failure of a link, say $e_{ll'} \in E$. As seen from figure 2, the total number of paths which use $e_{ll'}$ (as direct or two-hop path link) is equal to $2(n-2) + 1 = 2n - 3$.

The LP heuristic.

Step 1. Initialize (\bar{y}_{ij}) to zero and let $F = E$.

Step 2. Solve the linear programming problem (P0).

Update (\bar{y}_{ij}) as $(\bar{y}_{ij}) + (y_{ij})$.

Step 3. Obtain the matrix of total flow as $(R_{ij}) = (\sum_{s \in S} r_{ij}^s)$.

Step 4. Pick an origin–destination pair from F in non-descending order of R_{ij} values, say $e_{ij} \in F$ and solve (MP) for e_{ij} .

Update (\bar{y}_{ij}) as $(\bar{y}_{ij}) + (y_{ij})$, delete e_{ij} from F .

Step 5. Repeat step 4 until $F = \emptyset$.

On termination (\bar{y}_{ij}) is a feasible solution to problem (MS).

2.2. A simple heuristic for unbounded U_{ij}^s

The following very simple heuristic for the case of unbounded U_{ij}^s turns out to be surprisingly effective, as shown below, especially for the case when $c_{ij} = 1$ for all $i < j$.

The simple heuristic. Let

$$y_{ij} = \sum_{s \in S} r_{ij}^s + \frac{1}{n-2} \sum_{s \in S} \max_{k \neq i, j} (r_{ik}^s, r_{jk}^s).$$

Since there are $n-2$ two-hop paths between any two nodes it is clear that the above heuristic provides a feasible solution. As it does not take the costs into account we would expect it perform best when all costs are equal. This is confirmed numerically, below.

For each s and i , calculate the largest and second largest r_{ik}^s and their indices k , in $O(N_s n^2)$. Then the y_{ij} are computed in $O(N_s)$ each. So the computational complexity of the simple heuristic is $O(N_s n^2)$.

3. Numerical results

To compare the performance of the three mathematical methods, two groups of sample networks were tested. Both groups have the number of nodes varying between 4 and 12 and the number of service types fixed at 3. The unit capacity cost for each edge in group A is fixed at 1, whereas in group B the unit capacity cost for each edge was randomly selected from uniformly distributed integer values from 50 to 99. The AMPL (A Modelling Language for Mathematical Programming) optimisation package was used

for implementing the codes for each of the three solution methods. CPLEX provided the solution engine. Traffic demand for each link was similarly created using uniformly distributed integer values from 0 to 49. For practical reasons all upperbound U_{ij}^s values were set to a large number 1000, as insufficiently large U_{ij}^s values may have resulted in an infeasible problem. All the tests were run on a SUNW, SPARCstation-10 with 125 MHz processor and 128 MB RAM. Tables 1 and 2 show the total network link costs obtained using the three different methods for test group A and B, respectively. Note that all the computational results reported here are obtained from single problems, that is, for each node size only one randomly generated instance was used.

Table 3 gives an approximate count of the time taken by the processor to produce a solution. (Data in this table was obtained from running test group B problems. The computation time taken for running test group A has a similar trend to what is presented in this table.) Both the LP and simple heuristics gave more expensive solutions than the exact optimal solution obtained from the large LP problem, MS. The LP heuristic produces, in general, a solution which is closer to the optimal. The simple heuristic is by far the fastest and performs quite well especially when the costs of all links are constant.

Table 1

Total network link cost obtained from the three different methods (number of service types is fixed at 3, link cost = 1).

No. of nodes	Big-LP	LP heuristic	Simple heuristic
4	694	727.5 (4.82%)	779 (12.25%)
5	1039.5	1100 (5.82%)	1166 (12.17%)
6	1395.33	1573 (12.73%)	1558 (11.68%)
7	1736.4	1874 (7.92%)	1958.6 (12.80%)
8	2302.5	2493.82 (8.31%)	2561.83 (11.26%)
9	2906.29	3073.83 (5.76%)	3156.86 (8.62%)
10	3555.44	3793.91 (6.71%)	3900.12 (9.70%)
11	4391.32	4722.43 (7.54%)	4722 (7.53%)
12	out of memory	out of memory	5677 (N/A)

Table 2

Total network link cost obtained from the three different methods (number of service types is fixed at 3, link cost $\in [50,99]$).

No. of nodes	Big-LP	LP heuristic	Simple heuristic
4	54475	58187 (6.81%)	57648.5 (5.83%)
5	70788	81314 (14.87%)	80799.3 (14.14%)
6	93301.8	104092 (11.56%)	108136 (15.90%)
7	129693	143074 (10.32%)	147906 (14.04%)
8	163489	185203 (13.28%)	186276 (13.94%)
9	210717	232635 (10.40%)	233221 (10.68%)
10	266121	286381 (7.61%)	291157 (9.41%)
11	320358	346669 (8.21%)	351527 (9.73%)
12	out of memory	out of memory	438620 (N/A)

Table 3
Computation time comparison.

No. of nodes	Big-LP	LP heuristic	Simple heuristic
4	1 sec	1 sec	< 1 sec
5	1 sec	1 sec	< 1 sec
6	2 sec	2 sec	< 1 sec
7	9 sec	8 sec	< 1 sec
8	61 sec	15 sec	< 1 sec
9	6 min	28 sec	< 1 sec
10	27 min	60 sec	< 1 sec
11	92 min	105 sec	< 1 sec

However, if the link costs vary from link to link then the error will be greater for the simple heuristic since it does not take link costs into account, and thus it does not avoid using the links with greater costs.

4. Conclusions

We have formulated an optimal solution model for the problem of multi-service network design for the non-failure case and extended this for the case, where the network is subject to single link failure. Then we provided an LP heuristic for solving this problem. A simple heuristic which is $O(N_s n^2)$ was shown to be effective, especially for the case of equal link costs.

References

- Ash, G.R., R.H. Cardwell, and R.P. Murray. (1981). "Design and Optimization of Networks with Dynamic Routing." *Bell System Tech. Journal* 60(8), 1787–1820.
- Cameron, W.H., T. Regnier, P. Galloy, and A.M. Savoie. (1983). "Dynamic Routing for Intercity Telephone Networks." In *10th International Teletraffic Congress*, Paper 3.2-3.
- Gibbens, R.J. and P.B. Key. (1988). "Dynamic Alternative Routing-Modelling and Behaviour." In *12th International Teletraffic Congress*, Paper 3.4A-3.
- Gibbens, R.J. and F.P. Kelly. (1990). "Dynamic Routing in Fully Connected Networks." *IMA Journal of Math. Control & Infor.* 7, 77–111.
- Medhi, D. (1991). "Network Dimensioning for a Multi-Service Integrated Digital Network with Dynamic Routing." In *Proceedings of ACM/IEEE-CS Symposium on Applied Computing (SAC'91)*, pp. 30–36.
- Ouveysi, I. and Y.K. Tham. (1994). "Survivable Network Design for Volatile Traffic Scenarios." In *Proceedings of Australian Telecommunication Networks & Applications Conference*, pp. 851–856.